

Obtaining Error-Minimizing Estimates and Universal Entry-Wise Error Bounds for Low-Rank Matrix Completion

Franz J. Király*

Louis Theran[†]

Abstract

We propose a general framework for reconstructing and denoising single entries of incomplete and noisy entries. We describe: effective algorithms for deciding if an entry can be reconstructed and, if so, for reconstructing and denoising it; and a priori bounds on the error of each entry, individually. In the noiseless case our algorithm is exact. For rank-one matrices, the new algorithm is fast, admits a highly-parallel implementation, and produces an error minimizing estimate that is qualitatively close to our theoretical and the state-of-the-art Nuclear Norm and OptSpace methods.

1. Introduction

Matrix Completion is the task to reconstruct low-rank matrices from a subset of its entries and occurs naturally in many practically relevant problems, such as missing feature imputation, multi-task learning (Argyriou et al., 2008), transductive learning (Goldberg et al., 2010), or collaborative filtering and link prediction (Srebro et al., 2005; Acar et al., 2009; Menon and Elkan, 2011).

Almost all known methods performing matrix completion are optimization methods such as the max-norm and nuclear norm heuristics (Srebro et al., 2005; Candès and Recht, 2009; Tomioka et al., 2010), or OptSpace (Keshavan et al., 2010), to name a few amongst many.

These methods have in common that in general (a) they reconstruct the whole matrix and (b) error bounds are given for all of the matrix, not single entries. These two properties of existing methods are in particular unsatisfactory¹ in the scenario when one is interested only in predicting (resp. imputing) one single missing entry or a set of interesting missing entries instead of all - which is for real data a more natural task than imputing all missing entries, in particular in the presence of large scale data (resp. big data).

Indeed the design of such a method is not only desirable but also feasible, as the results of Király et al. (2012) suggest by relating algebraic combinatorial properties and the low-rank setting to the reconstructability of the data. Namely, the authors provide algorithms which can decide for one entry if it can be - in principle - reconstructed or not, thus yielding a statement of trustability for the output of any algorithm².

*Machine Learning Group, TU-Berlin, franz.j.kiraly@tu-berlin.de

[†]Discrete Geometry Group, FU Berlin, theran@math.fu-berlin.de

¹While the existing methods may be applied to a submatrix, it is always at the cost of accuracy if the data is sparse, and they do not yield statements on single entries.

²The authors also provide an algorithm for reconstructing some missing entries in the arbitrary rank case, but without obtaining global or entry-wise error bounds, or a strategy to reconstruct all reconstructible entries.

In this paper, we demonstrate the first time how algebraic combinatorial techniques, combined with stochastic error minimization, can be applied to (a) reconstruct single missing entries of a matrix and (b) provide lower variance bounds for the error of any algorithm resp. estimator for that particular entry - where the error bound can be obtained without actually reconstructing the entry in question. In detail, our contributions include:

- the construction of a variance-minimal and unbiased estimator for any fixed missing entry of a rank-one-matrix, under the assumption of known noise variances
- an explicit form for the variance of that estimator, being a lower bound for the variance of any unbiased estimation of any fixed missing entry and thus yielding a quantitative measure on the trustability of that entry reconstructed from any algorithm
- the description of a strategy to generalize the above to any rank
- comparison of the estimator with two state-of-the-art optimization algorithms (OptSpace and nuclear norm), and error assessment of the three matrix completion methods with the variance bound

Note that most of the methods and algorithms presented in this paper restrict to rank one. This is not, however, inherent in the overall scheme, which is general. We depend on rank one only in the sense that we understand the combinatorial-algebraic structure of rank-one-matrix completion exactly, whereas the behavior in higher rank is not yet as well understood. Nonetheless, it is, in principle accessible, and, once available will can be “plugged in” to the results here without changing the complexity much.

2. The Algebraic Combinatorics of Matrix Completion

2.1. A review of known facts In Király et al. (2012), an intricate connection between the algebraic combinatorial structure, asymptotics of graphs and analytical reconstruction bounds has been exposed. We will refine some of the theoretical concepts presented in that paper which will allow us to construct the entry-wise estimator.

Definition 2.1 *An matrix $M \in \{0, 1\}^{m \times n}$ is called mask. If A is a partially known matrix, then the mask of A is the mask which has 1-s in exactly the positions which are known in A ; and 0-s otherwise.*

Definition 2.2 *Let M be an $(m \times n)$ mask. We will call the unique bipartite graph $G(M)$ which has M as bipartite adjacency matrix the completion graph of M . We will refer to the m vertices of $G(M)$ corresponding to the rows of M as blue vertices, and to the n vertices of $G(M)$ corresponding to the columns as red vertices. If $e = (i, j)$ is an edge in $K_{m,n}$ (where $K_{m,n}$ is the complete bipartite graph with m blue and n red vertices), we will also write A_e instead of A_{ij} and for any $(m \times n)$ matrix A .*

A fundamental result, (Király et al., 2012, Theorem 2.3.5), says that identifiability and reconstructability are, up to a null set, graph properties.

Theorem 2.3 *Let A be a generic³ and partially known $(m \times n)$ matrix of rank r , let M be the mask of A , let i, j be integers. Whether A_{ij} is reconstructible (uniquely, or up to finite choice) depends only on M and the true rank r ; in particular, it does not depend on the true A .*

³In particular, if A is sampled from a continuous density, then the set of non-generic A is a null set.

For rank one, as opposed to higher rank, the set of reconstructible entries is easily obtainable from $G(M)$ by combinatorial means:

Theorem 2.4 ((Király et al., 2012, Theorem 2.5.36 (i))) *Let $G \subseteq K_{m,n}$ be the completion graph of a partially known $(m \times n)$ matrix A . Then the set of uniquely reconstructible entries of A is exactly the set A_e , with e in the transitive closure of G . In particular, all of A is reconstructible if and only if G is connected.*

2.2. Reconstruction on the transitive closure We extend Theorem 2.4's theoretical reconstruction guarantee by describing an explicit, algebraic algorithm for actually doing the reconstruction. This algorithm will be the basis of an entry-wise, variance-optimal estimator in the noisy case. In any rank, such a reconstruction rule can be obtained by exposing equations which explicitly give known and unknown entries in terms of only known entries due to the fact that the set of low-rank matrices is an irreducible variety (the common vanishing locus of finitely many polynomial equations). We are able to derive the reconstruction equations for rank one.

Definition 2.5 *Let $P \subseteq K_{m,n}$ (resp. $C \subseteq K_{m,n}$) be a path (resp. cycle), with a fixed start and end (resp. traversal order). We will denote by $E^+(P)$ be the set of edges in P (resp. $E^+(C)$ and C) traversed from blue vertex to a red one, and by $E^-(P)$ the set of edges traversed from a red vertex to a blue one⁴. From now on, when we speak of “oriented paths” or “oriented cycles”, we mean with this sign convention and some fixed traversal order.*

Let $A = A_{ij}$ be a $(m \times n)$ matrix of rank 1, and identify the entries A_{ij} with the edges of $K_{m,n}$. For an oriented cycle C , we define the polynomials

$$P_C(A) = \prod_{e \in E^+(C)} A_e - \prod_{e \in E^-(C)} A_e, \quad \text{and}$$

$$L_C(A) = \sum_{e \in E^+(C)} \log A_e - \sum_{e \in E^-(C)} \log A_e,$$

where for negative entries of A , we fix a branch of the complex logarithm.

Theorem 2.6 *Let $A = A_{ij}$ be a generic $(m \times n)$ matrix of rank 1. Let $C \subseteq K_{m,n}$ be an oriented cycle. Then, $P_C(A) = L_C(A) = 0$.*

Proof: The determinantal ideal of rank one is a binomial ideal generated by the (2×2) minors of A (where entries of A are considered as variables). The minor equations are exactly $P_C(A)$, where C is an elementary oriented four-cycle; if C is an elementary 4-cycle, denote its edges by $a(C)$, $b(C)$, $c(C)$, $d(C)$, with $E^+(C) = \{a(C), d(C)\}$. Let \mathcal{C} be the collection of the elementary 4-cycles, and define $L_{\mathcal{C}}(A) = \{L_C(A) : C \in \mathcal{C}\}$ and $P_{\mathcal{C}}(A) = \{P_C(A) : C \in \mathcal{C}\}$.

By sending the term $\log A_e$ to a formal variable x_e , we see that the free \mathbb{Z} -group generated by the $L_C(A)$ is isomorphic to $H_1(K_{m,n}, \mathbb{Z})$. With this equivalence, it is straightforward that, for any oriented cycle D , $L_D(A)$ lies in the \mathbb{Z} -span of elements of $L_{\mathcal{C}}(A)$ and, therefore, formally,

$$L_D(A) = \sum_{C \in \mathcal{C}} \alpha_C \cdot L_C(A)$$

⁴This is equivalent to fixing the orientation of $K_{m,n}$ that directs all edges from blue to red, and then taking $E^+(P)$ to be the set of edges traversed forwards and $E^-(P)$ the set of edges traversed backwards. This convention is convenient notationally, but any initial orientation of $K_{m,n}$ will give us the same result.

with the $\alpha_c \in \mathbb{Z}$. Thus $L_D(\cdot)$ vanishes when A is rank one, since the r.h.s. does. Exponentiating, we see that

$$\left(\prod_{e \in E^+(D)} A_e \right) \left(\prod_{e \in E^-(D)} A_e \right)^{-1} = \prod_{C \in \mathcal{C}} \left(A_{a(C)} A_{d(C)} A_{b(C)}^{-1} A_{c(C)}^{-1} \right)^{\alpha_C}$$

If A is generic and rank one, the r.h.s. evaluates to one, implying that $P_D(A)$ vanishes. \square

Corollary 2.7 *Let $A = A_{ij}$ be a $(m \times n)$ matrix of rank 1. Let v, w be two vertices in $K_{m,n}$. Let P, Q be two oriented paths in $K_{m,n}$ starting at v and ending at w . Then, for all A , it holds that $L_P(A) = L_Q(A)$.*

Remark 2.8 *It is possible to prove that the set of P_C forms the set of polynomials vanishing on the entries of A which is minimal with respect to certain properties. Namely, the P_C form a universal Gröbner basis for the determinantal ideal of rank 1, which implies the converse of Theorem 2.6. From this, one can deduce that the estimators presented in section 3.2 are variance-minimal amongst all unbiased ones.*

3. A Combinatorial Algebraic Estimate for Missing Entries and Their Error

In this section, we will construct an estimator for matrix completion which (a) is able to complete single missing entries and (b) gives universal error estimates for that entry that are independent of the reconstruction algorithm.

3.1. The sampling model In all of the following, we will assume that the observations arise from the following sampling process:

Assumption 3.1 *There is an unknown fixed, rank one, matrix A which is generic, and an $(m \times n)$ mask $M \in \{0, 1\}^{m \times n}$ which is known. There is a (stochastic) noise matrix $\mathcal{E} \in \mathbb{R}^{m \times n}$ whose entries are uncorrelated and which is multiplicatively centered with finite variance, non-zero⁵ variance; i.e., $\mathbb{E}(\log \mathcal{E}_{ij}) = 0$ and $0 < \text{Var}(\log \mathcal{E}_{ij}) < \infty$ for all i and j .*

The observed data is the matrix $A \circ M \circ \mathcal{E} = \Omega(A \circ \mathcal{E})$, where \circ denotes the Hadamard (i.e., component-wise) product. That is, the observation is a matrix with entries $A_{ij} \cdot M_{ij} \cdot \mathcal{E}_{ij}$.

The assumption of multiplicative noise is a necessary precaution in order for the presented estimator (and in fact, any estimator) for the missing entries to have bounded variance, as shown in Example 3.2 below. This is not, in practice, a restriction since an infinitesimal additive error δA_{ij} on an entry of A is equivalent to an infinitesimal multiplicative error $\delta \log A_{ij} = \delta A_{ij} / A_{ij}$, and additive variances can be directly translated into multiplicative variances if the density function for the noise is known⁶. The previous observation implies that the multiplicative noise model is as powerful as any additive one that allows bounded variance estimates.

⁵The zero-variance case corresponds to exact reconstruction, which is handled already by Theorem 2.4.

⁶The multiplicative noise assumption causes the observed entries and the true entries to have the same sign. The change of sign can be modeled by adding another multiplicative binary random variable in the model which takes values ± 1 ; this adds an independent combinatorial problem for the estimation of the sign which can be done by maximum likelihood. In order to keep the exposition short and easy, we did not include this into the exposition.

Example 3.2 Consider the rank one matrix

$$A = \begin{pmatrix} A_{11} & A_{21} \\ A_{12} & A_{22} \end{pmatrix}.$$

The unique equation between the entries is $A_{11}A_{22} = A_{12}A_{21}$. Solving for any entry will have another entry in the denominator, for example

$$A_{11} = \frac{A_{12}A_{21}}{A_{22}}.$$

Thus we get an estimator for A_{11} when substituting observed and noisy entries for A_{12}, A_{21}, A_{22} . When A_{22} approaches zero, the estimation error for A_{11} approaches infinity. In particular, if the density function of the error E_{22} of A_{22} is too dense around the value $-A_{22}$, then the estimate for A_{11} given by the equation will have unbounded variance. In such a case, one can show that no estimator for A_{11} has bounded variance.

3.2. Estimating entries and error bounds In this section, we construct the unbiased estimator for the entries of a rank-one-matrix with minimal variance. First, we define some notation to ease the exposition:

Notations 3.3 We will denote by $a_{ij} = \log A_{ij}$ and $\varepsilon_{ij} = \log \mathcal{E}_{ij}$ the logarithmic entries and noise. Thus, for some path P in $K_{m,n}$ we obtain

$$L_P(A) = \sum_{e \in E^+(P)} a_e - \sum_{e \in E^-(P)} a_e.$$

Denote by $b_{ij} = a_{ij} + \varepsilon_{ij}$ the logarithmic (observed) entries, and B the (incomplete) matrix which has the (observed) b_{ij} as entries. Denote by $\sigma_{ij} = \text{Var}(b_{ij}) = \text{Var}(\varepsilon_{ij})$.

The components of the estimator will be built from the L_P :

Lemma 3.4 Let $G = G(M)$ be the graph of the mask M . Let $x = (v, w) \in K_{m,n}$ be any edge with v red. Let P be an oriented path⁷ in $G(M)$ starting at v and ending at w . Then,

$$L_P(B) = \sum_{e \in E^+(P)} b_e - \sum_{e \in E^-(P)} b_e$$

is an unbiased estimator for a_x with variance

$$\text{Var}(L_P(B)) = \sum_{e \in P} \sigma_e.$$

Proof: By linearity of expectation and centeredness of ε_{ij} , it follows that

$$\mathbb{E}(L_P(B)) = \sum_{e \in E^+(P)} \mathbb{E}(b_e) - \sum_{e \in E^-(P)} \mathbb{E}(b_e),$$

⁷If $x \in G$, then P can also be the path consisting of the single edge e .

thus $L_P(B)$ is unbiased. Since the ε_e are uncorrelated, the b_e also are; thus, by Bienaymé's formula, we obtain

$$\text{Var}(L_P(B)) = \sum_{e \in E^+(P)} \text{Var}(b_e) + \sum_{e \in E^-(P)} \text{Var}(b_e),$$

and the statement follows from the definition of σ_e .

In the following, we will consider the following parametric estimator as a candidate for estimating a_e :

Notations 3.5 Fix an edge $x = (v, w) \in K_{m,n}$. Let \mathcal{P} be a basis for the set of all oriented paths starting at v and ending at w ⁸, and denote $\#\mathcal{P}$ by p . For $\alpha \in \mathbb{R}^p$, set

$$X(\alpha) = \sum_{P \in \mathcal{P}} \alpha_P L_P(B).$$

Furthermore, we will denote by $\mathbb{1}$ the n -vector of ones.

The following Lemma follows immediately from Lemma 3.4 and Theorem 2.6:

Lemma 3.6 $\mathbb{E}(X(\alpha)) = \mathbb{1}^\top \alpha \cdot b_x$; in particular, $X(\alpha)$ is an unbiased estimator for b_x if and only if $\mathbb{1}^\top \alpha = 1$.

We will now show that minimizing the variance of $X(\alpha)$ can be formulated as a quadratic program with coefficients entirely determined by a_x , the measurements b_e and the graph $G(M)$. In particular, we will expose an explicit formula for the α minimizing the variance. Before stating the theorem, we define a suitable kernel:

Definition 3.7 Let $e \in K_{m,n}$ be an edge. For an edge e and a path P , set $c_{e,P} = \pm 1$ if $e \in E^\pm(P)$ otherwise $c_{e,P} = 0$. Let $P, Q \in \mathcal{P}$ be any fixed oriented paths. Define the (weighted) path kernel $k : \mathcal{P} \times \mathcal{P} \rightarrow \mathbb{R}$ by

$$k(P, Q) = \sum_{e \in K_{m,n}} c_{e,P} \cdot c_{e,Q} \cdot \sigma_e.$$

Under our assumption that $\text{Var}(b_e) > 0$ for all $e \in K_{m,n}$, the path kernel is positive definite, since it is a sum of p independent positive semi-definite functions; in particular, its kernel matrix has full rank. Here is the variance-minimizing unbiased estimator:

Proposition 3.8 Let $x = (s, t)$ be a pair of vertices, and \mathcal{P} a basis for the s - t path space in G with p elements. Let Σ be the $p \times p$ kernel matrix of the path kernel with respect to the basis \mathcal{P} . For any $\alpha \in \mathbb{R}^p$,

$$\text{Var}(X(\alpha)) = \alpha^\top \Sigma \alpha.$$

Moreover, under the condition $\mathbb{1}^\top \alpha = 1$, the variance $\text{Var}(X(\alpha))$ is minimized by

$$\alpha = \left(\Sigma^{-1} \mathbb{1} \right) \left(\mathbb{1}^\top \Sigma^{-1} \mathbb{1} \right)^{-1}$$

⁸This is the set of words equal to the formal generators $x_{(v,w)}$ in the free abelian group generated by the x_e , subject to the relations $L_C = 0$ for all cycles C in $G \cup \{(v, w)\}$. Independence can be taken as linear independence of the coefficient vectors of the L_C .

Proof: By inserting definitions, we obtain

$$\begin{aligned} X(\alpha) &= \sum_{P \in \mathcal{P}} \alpha_P L_P(B) \\ &= \sum_{P \in \mathcal{P}} \alpha_P \sum_{e \in K_{m,n}} c_{e,P} b_e. \end{aligned}$$

Writing $b = (b_e) \in \mathbb{R}^{mn}$ as vectors, and $C = (c_{e,p}) \in \mathbb{R}^{p \times mn}$ as matrices, we obtain

$$X(\alpha) = b^\top C \alpha.$$

By using that $\text{Var}(\lambda \cdot) = \lambda^2 \text{Var}(\cdot)$ for any scalar λ , and independence of the b_e , an elementary calculation yields

$$\text{Var}(X(\alpha)) = \alpha^\top \Sigma \alpha$$

In order to determine the minimum of the variance in α , consider the Lagrangian

$$L(\alpha, \lambda) = \alpha^\top \Sigma \alpha + \lambda \left(1 - \sum_{P \in \mathcal{P}} \alpha_P \right),$$

where the slack term models the condition $\ell(\alpha) = 1$. An elementary calculation yields

$$\frac{\partial L}{\partial \alpha} = 2\Sigma \alpha - \lambda \mathbb{1}$$

where $\mathbb{1}$ is the vector of ones. Due to positive definiteness of Σ the function $\text{Var}(X(\alpha))$ is convex, thus $\alpha = \Sigma^{-1} \mathbb{1} / \mathbb{1}^\top \Sigma^{-1} \mathbb{1}$ will be the unique α minimizing the variance while satisfying $\mathbb{1}^\top \alpha = 1$. \square

Remark 3.9 *The above setup works in wider generality: (i) if $\text{Var}(b_e) = 0$ is allowed and there is an s - t path of all zero variance edges, the path kernel becomes positive semi-definite; (ii) similarly if \mathcal{P} is replaced with any set of paths at all, the same may occur. In both cases, we may replace Σ^{-1} with the Moore-Penrose pseudo-inverse and the proposition still holds: (i) reduces to the exact reconstruction case of Theorem 2.4; (ii) produces the optimal estimator with respect to \mathcal{P} , which is optimal provided that \mathcal{P} is spanning, and adding paths to \mathcal{P} does not make the estimate worse.*

3.3. Rank 2 and higher An estimator for rank 2 and higher, together with a variance analysis, can be constructed similarly once all polynomials known which relate the entries under each other. The main difficulty lies in the fact that these polynomials are not parameterized by cycles anymore, but specific subgraphs of $G(M)$, see (Király et al., 2012, Section 2.5). Were these polynomials known, an estimator similar to $X(\alpha)$ as in Notation 3.5 could be constructed, and a subsequent variance (resp. perturbation) analysis performed.

3.4. The algorithms In this section, we describe the two main algorithms which calculate the variance-minimizing estimate \hat{A}_{ij} for any fixed entry A_{ij} of an $(m \times n)$ matrix A , which is observed with noise, and the variance bound for the estimate \hat{A}_{ij} . It is important to note that A_{ij} does not necessarily need to be an entry which is missing in the observation, it can also be any entry which

Algorithm 1 Calculates path kernel Σ and α .

Input: index (i, j) , an $(m \times n)$ mask M , variances σ .

Output: path matrix C , path kernel Σ and minimizer α .

- 1: Find a linearly independent set of paths \mathcal{P} in the graph $G(M)$, starting from i and ending at j .
 - 2: Determine the matrix $C = (c_{e,P})$ with $e \in G(M), P \in \mathcal{P}$; set $c_{e,P} = \pm 1$ if $e \in E^\pm(P)$, otherwise $c_{e,P} = 0$.
 - 3: Define a diagonal matrix $S = \text{diag}(\sigma)$, with $S_{ee} = \sigma_e$ for $e \in G(M)$.
 - 4: Compute the kernel matrix $\Sigma = C^\top S C$.
 - 5: Calculate $\alpha = \Sigma^{-1} \mathbb{1} / \|\Sigma^{-1} \mathbb{1}\|_1$.
 - 6: Output C, Σ and α .
-

has been observed. In the latter case, Algorithm 3 will give an improved estimate of the observed entry, and Algorithm 4 will give the trustworthiness bound on this estimate.

Since the the path matrix C , the path kernel matrix Σ , and the optimal α is required for both, we first describe Algorithm 1 which determines those. The steps of the algorithm follow the exposition in section 3.2, correctness follows from the statements presented there. The only task in Algorithm 1 that isn't straightforward is the computation of a linearly independent set of paths in step 1. We can do this time linear in the number of observed entries in the mask M with the following method. To keep the notational manageable, we will conflate formal sums of the x_e , cycles in $H_1(G, \mathbb{Z})$ and their representations as vectors in \mathbb{R}^{mn} , since there is no chance of confusion. We prove the correctness of Algorithm 2.

Algorithm 2 Calculates a basis \mathcal{P} of the path space.

Input: index (i, j) , an $(m \times n)$ mask M .

Output: a basis \mathcal{P} for the space of oriented i - j paths.

- 1: If (i, j) is not an edge of M , and i and j are in different connected components, then \mathcal{P} is empty. Output \emptyset .
 - 2: Otherwise, if (i, j) is not an edge, of M , add a “dummy” copy.
 - 3: Compute a spanning forest F of M that does not contain (i, j) , if possible.
 - 4: For each edge $e \in M \setminus F$, compute the fundamental cycle C_e of e in F .
 - 5: If (i, j) is an edge in M , output $\{-x_{(i,j)}\} \cup \{C_e - x_{(i,j)} : e \in M \setminus F\}$.
 - 6: Otherwise, let $P_{(i,j)} = C_{(i,j)} - x_{(i,j)}$. Output $\{C_e - P_{(i,j)} : e \in M \setminus (F \cup \{(i, j)\})\}$.
-

Algorithms 3 and 4 then can make use of the calculated C, α, Σ to determine an estimate for any entry A_{ij} and its minimum variance bound. The algorithms follow the exposition in Section 3.2, from where correctness follows; Algorithm 3 additionally provides treatment for the sign of the entries.

Note that even if observations are not available, Algorithm 4 can be used to obtain the variance bound. The variance bound is relative, due to its multiplicativity, and can be used to approximate absolute bounds when any reconstruction estimate \hat{A}_{ij} is available - which does not necessarily need to be the one from Algorithm 3, but can be the estimation result of any reconstruction. Namely, if $\hat{\sigma}_{ij}$ is the estimated variance of the log, we obtain an upper confidence bound (resp. deviation) bound $\hat{A}_{ij} \cdot \exp(\sqrt{\hat{\sigma}_{ij}})$ for \hat{A}_{ij} , and a lower confidence bound (resp. de-

Algorithm 3 Estimates the entry a_{ij} .

Input: index (i, j) , an $(m \times n)$ mask M , log-variances σ , the partially observed and noisy matrix B .

Output: The variance-minimizing estimate for A_{ij} .

- 1: Calculate C and α with Algorithm 1.
 - 2: Store B as a vector $b = (\log |B_e|)$ and a sign vector $s = (\text{sgn } B_e)$ with $e \in G(M)$.
 - 3: Calculate $\hat{A}_{ij} = \pm \exp(b^\top C \alpha)$. The sign is $+$ if each column of $s^\top |C|$ ($| \cdot |$ component-wise) contains an odd number of entries -1 , else $-$.
 - 4: Return \hat{A}_{ij} .
-

Algorithm 4 Determines the variance of the entry $\log(A_{ij})$.

Input: index (i, j) , an $(m \times n)$ mask M , log-variances σ .

Output: The variance lower bound for $\log(A_{ij})$.

- 1: Calculate Σ and α with Algorithm 1.
 - 2: Return $\alpha^\top \Sigma \alpha$.
-

viation) bound $\hat{A}_{ij} \cdot \exp(-\sqrt{\hat{\sigma}_{ij}})$, corresponding to the log-confidence $\log \hat{A}_{ij} \pm \sqrt{\hat{\sigma}_{ij}}$. Also note that if A_{ij} is not reconstructible from the mask M (i.e., if the edge (i, j) is not in the transitive closure of $G(M)$, see Theorem 2.4), then the deviation bounds will be infinite.

4. Experiments

4.1. Universal error estimates For three different masks, we calculated the predicted minimum variance for each entry of the mask. The multiplicative noise was assumed to be $\sigma_e = 1$ for each entry. Figure 1 shows the predicted a-priori minimum variances for each of the masks. Notice how the structure of the mask affects the expected error; known entries generally have least variance, while it is interesting to note that in general it is less than the starting variance of 1. I.e., tracking back through the paths can be successfully used even to denoise known entries. The particular structure of the mask is mirrored in the pattern of the predicted errors; a diffuse mask gives a similar error on each missing entry, while the more structured masks have structured error which is determined by combinatorial properties of the completion graph and the paths therein.

4.2. Influence of noise level We generated 10 random mask of size 50×50 with 200 entries sampled uniformly and a random (50×50) matrix of rank one. The multiplicative noise was chosen entry-wise independent, with variance $\sigma_i = (i - 1)/10$ for each entry. Figure 2(a) compares the Mean Squared Error (MSE) for three algorithms: Nuclear Norm (using the implementation Tomioka et al. (2010)), OptSpace (Keshavan et al., 2010), and Algorithm 3. It can be seen that on this particular mask, Algorithm 3 is competitive with the other methods and even outperforms them for low noise.

4.3. Prediction of estimation errors The data are the same as in Section 4.2, as are the compared algorithm. Figure 2(b) compares the error of each of the methods with the variance predicted by Algorithm 4 each time the noise level changed. The figure shows that for any of

the algorithms, the mean of the actual error increases with the predicted error, showing that the error estimate is useful for a-priori prediction of the actual error - independently of the particular algorithm. Note that by construction of the data this statement holds in particular for entry-wise predictions. Furthermore, in quantitative comparison Algorithm 4 also outperforms the other two in each of the bins.

5. Conclusion

In this paper, we have introduced an algebraic combinatorics based method for reconstructing and denoising single entries of an incomplete and noisy matrix, and for calculating confidence bounds of single entry estimations for arbitrary algorithms. We have evaluated these methods against state-of-the art matrix completion methods. The results of section 4 show that our reconstruction method is competitive and that - for the first time - our variance estimate provides a reliable prediction of the error on each single entry which is an a-priori estimate, i.e., depending only on the noise model and the position of the known entries. Furthermore, our method allows to obtain the reconstruction and the error estimate for a single entry which existing methods are not capable of, possibly using only a small subset of neighboring entries - a property which makes our method unique and particularly attractive for application to large scale data. We thus argue that the investigation of the algebraic combinatorial properties of matrix completion, in particular in rank 2 and higher where these are not yet completely understood, is crucial for the future understanding and practical treatment of big data.

References

- E. Acar, D.M. Dunlavy, and T.G. Kolda. Link prediction on evolving data using matrix and tensor factorizations. In *Data Mining Workshops, 2009. ICDMW'09. IEEE International Conference on*, pages 262–269. IEEE, 2009.
- A. Argyriou, C. A. Micchelli, M. Pontil, and Y. Ying. A spectral regularization framework for multi-task structure learning. In J.C. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in NIPS 20*, pages 25–32. MIT Press, Cambridge, MA, 2008.
- Emmanuel J. Candès and Benjamin Recht. Exact matrix completion via convex optimization. *Found. Comput. Math.*, 9(6):717–772, 2009. ISSN 1615-3375. doi: 10.1007/s10208-009-9045-5. URL <http://dx.doi.org/10.1007/s10208-009-9045-5>.
- A. Goldberg, X. Zhu, B. Recht, J. Xu, and R. Nowak. Transduction with matrix completion: Three birds with one stone. In J. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R.S. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23*, pages 757–765. 2010.
- Raghunandan H. Keshavan, Andrea Montanari, and Sewoong Oh. Matrix completion from a few entries. *IEEE Trans. Inform. Theory*, 56(6):2980–2998, 2010. ISSN 0018-9448. doi: 10.1109/TIT.2010.2046205. URL <http://dx.doi.org/10.1109/TIT.2010.2046205>.
- Franz J. Király, Louis Theran, Ryota Tomioka, and Takeaki Uno. The algebraic combinatorial approach for low-rank matrix completion. Preprint, arXiv:1211.4116v3, 2012. URL <http://arxiv.org/abs/1211.4116>.

- A. Menon and C. Elkan. Link prediction via matrix factorization. *Machine Learning and Knowledge Discovery in Databases*, pages 437–452, 2011.
- N. Srebro, J. D. M. Rennie, and T. S. Jaakkola. Maximum-margin matrix factorization. In Lawrence K. Saul, Yair Weiss, and Léon Bottou, editors, *Advances in NIPS 17*, pages 1329–1336. MIT Press, Cambridge, MA, 2005.
- Ryota Tomioka, Kohei Hayashi, and Hisashi Kashima. On the extension of trace norm to tensors. In *NIPS Workshop on Tensors, Kernels, and Machine Learning*, 2010.

A. Correctness of Algorithm 2

We adopt the conventions of Section 2, so that G is a bipartite graph with m blue vertices, n red ones, and e edges oriented from blue to red. Recall the isomorphism, observed in the proof of Theorem 2.6 of the \mathbb{Z} -group of the polynomials $L_C(\cdot)$ and the oriented cycle space $H_1(G, \mathbb{Z})$.

Define $\beta_1(G) = e - n - m + c$ (the first Betti number of the graph). Some standard facts are that: (i) the rank of $H_1(G, \mathbb{Z})$ is $\beta_1(G)$; (ii) we can obtain a basis for $H_1(G, \mathbb{Z})$ consisting only of simple cycles by picking any spanning forest F of G and then using as basis elements the fundamental cycles C_e of the edges $e \in E \setminus F$. This justifies step 4.

Let (i, j) be an edge of G . Define an i - j to be the set of subgraphs such that, for generic rank one A , $L_P(A) = -x_{(i,j)}$. By Theorem 2.6, we can write these as \mathbb{Z} -linear combinations of $x_{(i,j)}$ and oriented cycles. From this, we see that the rank of the path space is $\beta_1(G) + 1$ and the graph theoretic identification of elements in the path space with subgraphs that have even degree at every vertex except i and j . Thus, if (i, j) is an edge of G , step 5 is justified, completing the proof of correctness in this case.

If (i, j) was not an edge, step 1 guarantees that the dummy copy of (i, j) that we added is not in the spanning tree computed in step 3. Thus, the element $P_{(i,j)} = C_{(i,j)} - x_{(i,j)}$ computed in step 6 is a simple path from i to j . The collection of elements generated in step 6 is independent by the same fact in $H_1(G \cup \{(i, j)\}, \mathbb{Z})$ and has rank $\beta_1(G) + 1$ and does not put a positive coefficient on the dummy generator $x_{(i,j)}$. \square

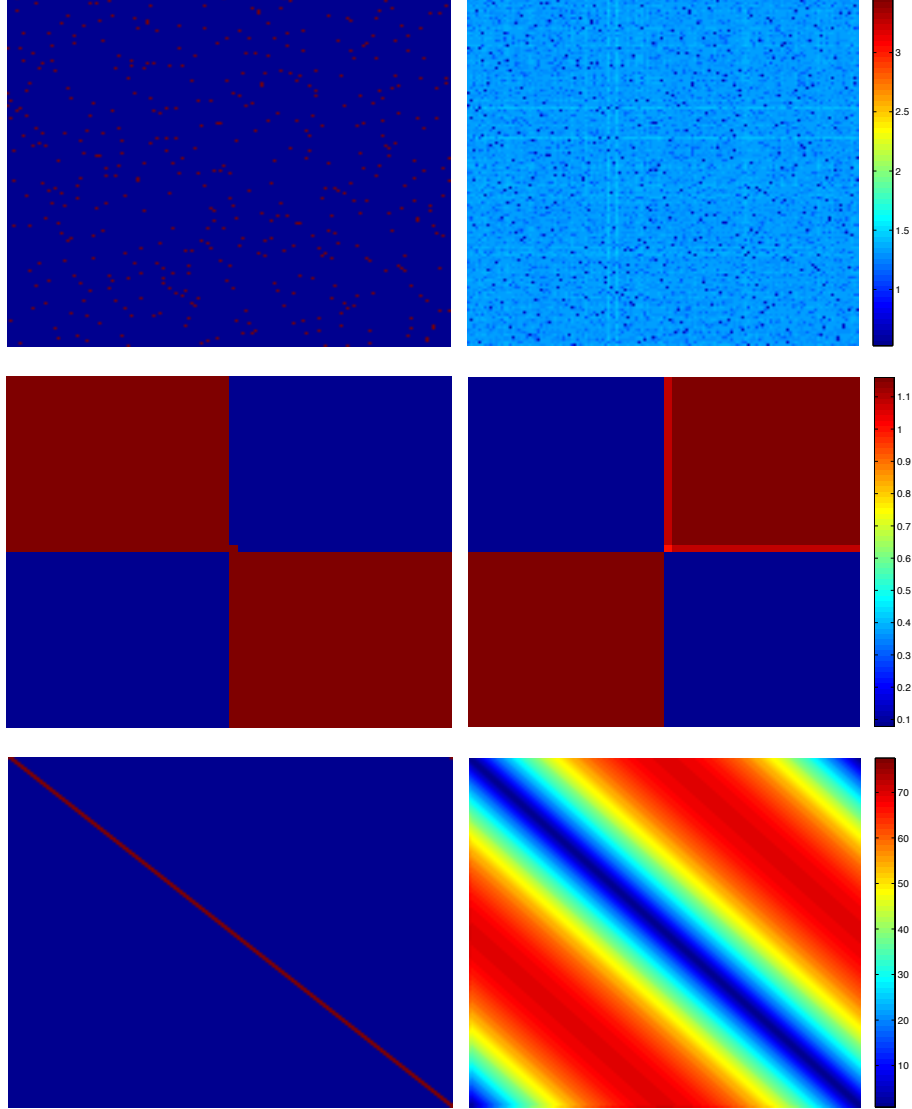


Figure 1: The figure shows three pairs of masks and predicted variances. A pair consists of two adjacent squares. The left half is the mask which is depicted by red/blue heatmap with red entries known and blue unknown. The right half is a multicolor heatmap with color scale, showing the predicted variance of the completion. Variances were calculated by our implementation of Algorithm 4.

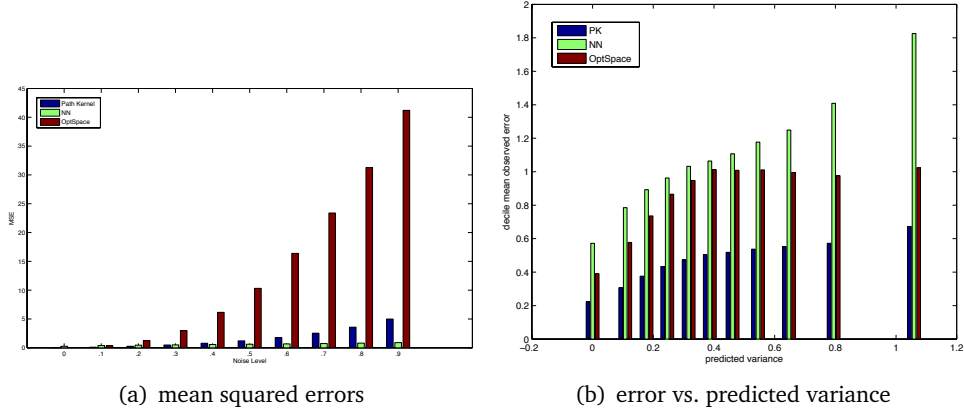


Figure 2: For 10 randomly chosen masks and 50×50 true matrix, matrix completions were performed with Nuclear Norm (green), OptSpace (red), and Algorithm 3 (blue) under multiplicative noise with variance increasing in increments of 0.1. For each completed entry, minimum variances were predicted by Algorithm 4. 2(a) shows the mean squared error of the three algorithms for each noise level, coded by the algorithms' respective colors. 2(b) shows a bin-plot of errors (y-axis) versus predicted variances (x-axis) for each of the three algorithms: for each completed entry, a pair (predicted error, true error) was calculated, predicted error being the predicted variance, and the actual prediction error measured as log abs of prediction minus log abs of true entry. Then, the points were binned into 11 bins with equal numbers of points. The figure shows the mean of the errors (second coordinate) of the value pairs with predicted variance (first coordinate) in each of the bins, the color corresponds to the particular algorithm; each group of bars is centered on the minimum value of the associated bin.